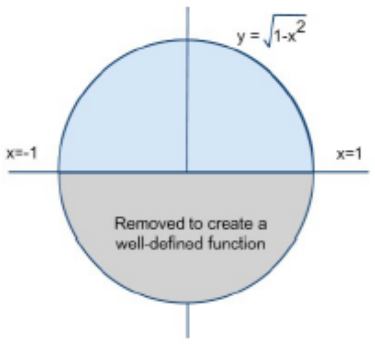
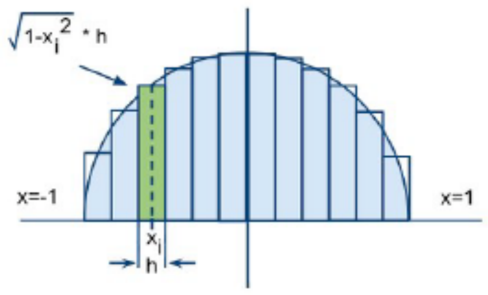
Homework 5

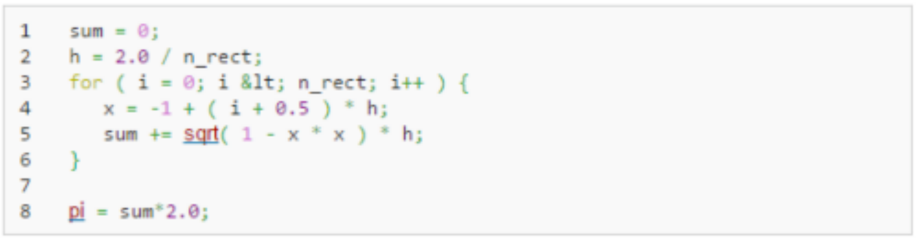
Total score: 20
Program language: C++
Submission: Compress source code (cpp) and analysis file (docx, pdf, txt) with tar.gz or zip, and submit it to Blackboard

Problem: Calculate Pi using numeric integration methods. Parallelize it using OpenMP and analyze the performance as below. Prepare a document (.txt, .pdf, or .docx) to analyze results.

Q1. The pseudocode shown below is the calculation of pi by the method of numeric integration. Complete the serial code. Refer the course slide “OpenMP_3”.

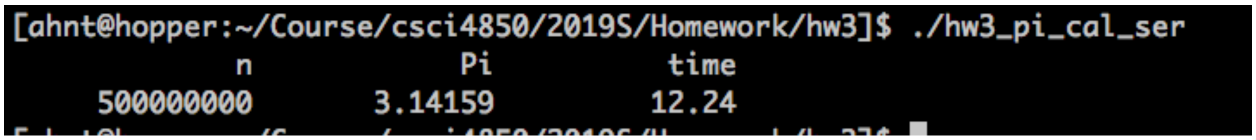






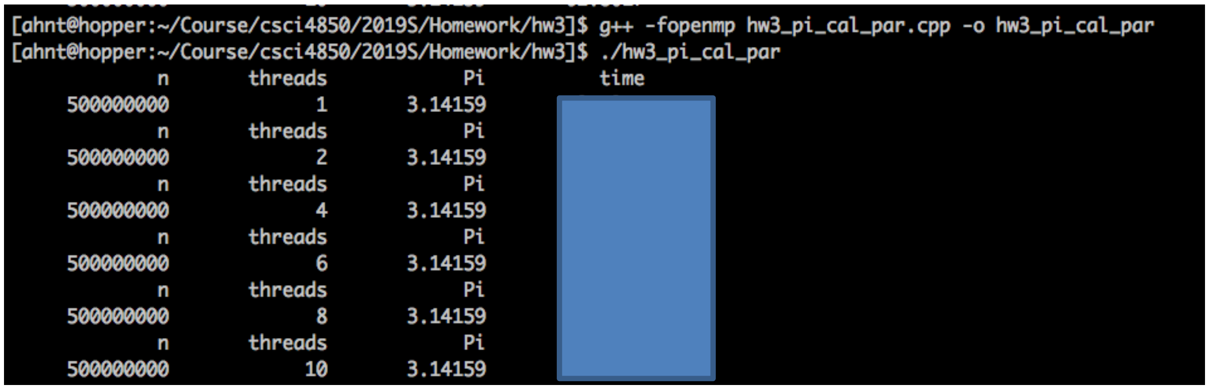
```
sum = 0;
h = 2.0 / n_rect;
for ( i = 0; i &lt; n_rect; i++ ) {
   x = -1 + ( i + 0.5 ) * h;
   sum += sqrt( 1 - x * x ) * h;
}

pi = sum*2.0;
```

- Set the number of rectangles to 500000000.
- Use clock_t begin = clock(); or any time function begin and end of the code to measure elapsed time.
- Generate output as below and put it in the analysis text format.



```
[ahnt@hopper:~/Course/csci4850/2019S/Homework/hw3]$ ./hw3_pi_cal_ser
            n              Pi           time
    500000000         3.14159          12.24
```

Q2. Parallelize it using "reduction" clause and test it with P=1, 2, 4, 6, 8, 10. Run it on hopper.slu.edu and report the results as below. Hope you get good speedup in this case.

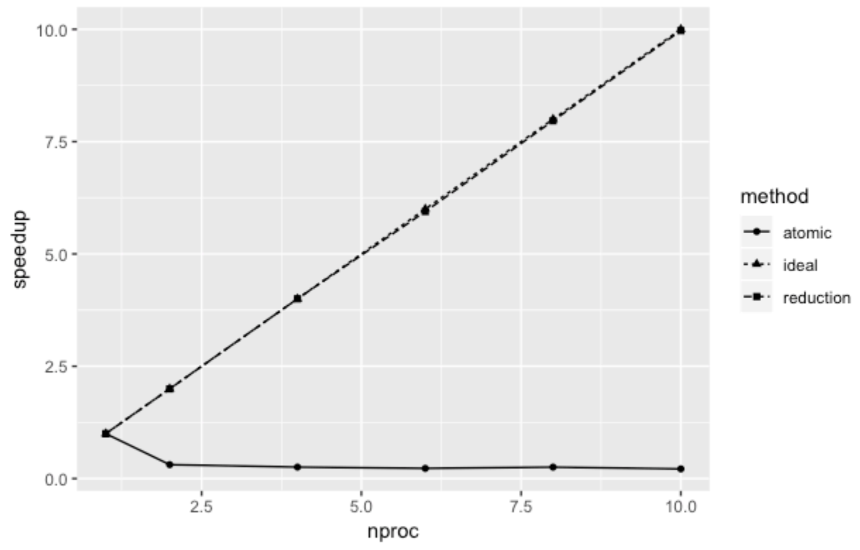


```
[ahnt@hopper:~/Course/csci4850/2019S/Homework/hw3]$ g++ -fopenmp hw3_pi_cal_par.cpp -o hw3_pi_cal_par
[ahnt@hopper:~/Course/csci4850/2019S/Homework/hw3]$ ./hw3_pi_cal_par
         n        threads           Pi            time
 500000000              1      3.14159        [redacted]
         n        threads           Pi
 500000000              2      3.14159
         n        threads           Pi
 500000000              4      3.14159
         n        threads           Pi
 500000000              6      3.14159
         n        threads           Pi
 500000000              8      3.14159
         n        threads           Pi
 500000000             10      3.14159
```

- You shoud use omp_get_wtime() to measure time.
- The calculated PI should be correct enough.

Q3. Change the "reduction" clause to use "atomic" directive. Then, test it with P=1, 2, 4, 6, 8. Report the results as previous.

Q4. Plot the speedup with three cases: ideal, reduction, and atomic as blow, then provide your comments why atomic is a bad idea for this case. I strongly recommend you to use R or MATLAB to plot it as below.


speedup
nproc
method
● atomic
▲ ideal
■ reduction